iOS malware can fake iPhone shut downs to snoop on camera, microphone

By **Bill Toulas (https://www.bleepingcomputer.com/author/bill-toulas/)**

January 5, 2022 09:54 AM 3

Researchers have developed a new technique that fakes a shutdown or reboot of iPhones, preventing malware from being removed and allowing hackers to secretly snoop on microphones and receive sensitive data via a live network connection.

Historically, when malware infects an iOS device, it can be removed simply by restarting the device, which clears the malware from memory.

However, this technique hooks the shutdown and reboot routines to prevent them from ever happening, allowing malware to achieve persistence as the device is never actually turned off.

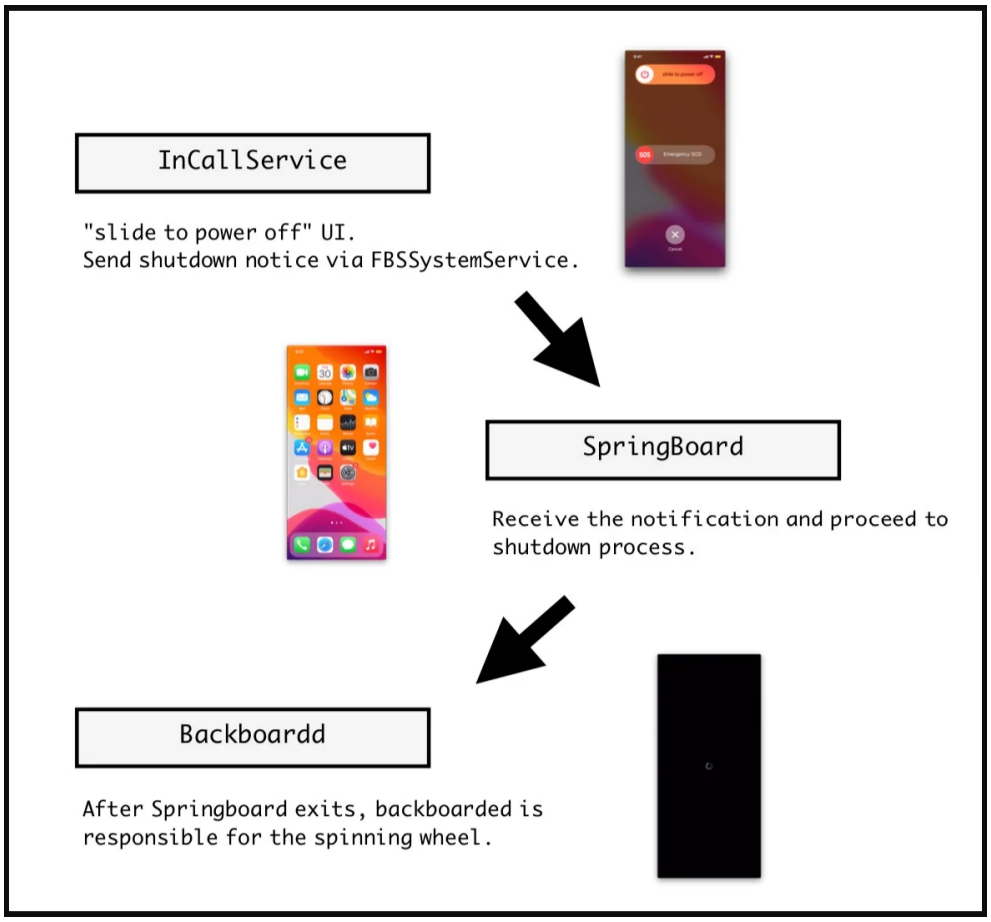
Because this attack, which the researchers call "NoReboot," does not exploit any flaws on the iOS and instead relies on human-level deception, it cannot be patched by Apple.

Simulating a convincing reboot

To restart the iPhone, one has to press and hold the power button and either volume button until the slider with the reboot option appears, and then wait for roughly 30 seconds for the action to complete.

When an iPhone is shut off, its screen naturally goes dark, the camera is turned off, 3D touch feedback does not respond to long presses, sounds from calls and notifications are muted, and all vibrations are absent.

Security researchers from ZecOps have developed a trojan PoC (https://github.com/ZecOps/public/tree/master/fake_shutdown_POC) (proof of concept) tool that can inject specially crafted code onto three iOS daemons to fake a shut down by disabling all the above indicators.



Hijacking three iOS daemons

Source: ZecOps

The trojan hijacks the shutdown event by hooking the signal sent to the "SpringBoard" (user interface interaction daemon).

Instead of the expected signal, the trojan will send a code that will force "SpingBoard" to exit, making the device non-responsive to user input. This is the perfect disguise in this case because devices that enter a shutdown state naturally no longer accept user inputs.

```
@interface BKSDefaults : NSObject
-(int)hideAppleLogoOnLaunch;
-(void)setHideAppleLogoOnLaunch:(int)arg1;
@end

void backboardd_hides_spinningWheel(){
    BKSDefaults *bks = objc_msgSend(objc_getClass("BKSDefaults"), @selector(localDefaults));
    [bks setHideAppleLogoOnLaunch:1];
}

int startMonitoring_powerOn = 0;
extern CFNotificationCenterRef CFNotificationCenterGetDistributedCenter(void);
void my_callback(CFNotificationCenterRef center, void *observer, CFNotificationName name, const void *object, CFDictionaryRef userInfo){
    backboardd_hides_spinningWheel();
    startMonitoring_powerOn = 1;
}
```

Code injected onto springboard

Source: ZecOps

Next, the "BackBoardd" daemon is commanded to display the spinning wheel that indicates the shutdown process is underway.

"BackBoardd" is another iOS daemon that logs physical button click and screen touch events with timestamps, so abusing it gives the trojan the power to know when the user attempts to "turn on" the phone.

By monitoring these actions, the user can be deceived to release the button earlier than they were supposed to, avoiding an actual forced restart.

ZecOps describes (https://blog.zecops.com/research/persistence-without-persistence-meet-the-ultimate-persistence-bug-noreboot/) the next step in the "NoReboot" attack as follows:

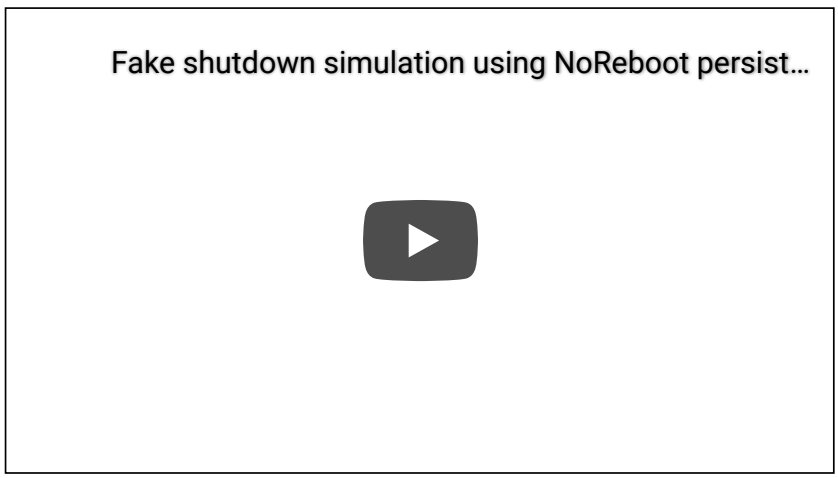
> *The file will unleash the SpringBoard and trigger a special code block in our injected dylib. What it does is to leverage local SSH access to gain root privilege, then we execute /bin/launchctl reboot userspace.*
>
> *This will exit all processes and restart the system without touching the kernel. The kernel remains patched. Hence malicious code won't have any problem continuing to run after this kind of reboot. The user will see the Apple Logo effect upon restarting.*
>
> *This is handled by backboardd as well. Upon launching the SpringBoard, the backboardd lets SpringBoard take over the screen.*

backboardd giving screen control back to springboard

Source: ZecOps

The user returns to a regular UI with all processes and services running as expected, with no indication that they just went through a simulated reboot.

Zecops has created a video showing the NoReboot technique in action, illustrating how it can easily trick anyone into thinking their device has been turned off.



Never trust that a device is fully turned off

Apple introduced a new feature in iOS 15, making it possible for users to locate their iPhones through 'Find My' even if they are powered off.

Apple didn't bother to explain how exactly that works, but researchers found (https://naehrdine.blogspot.com/2021/09/always-on-processor-magic-how-find-my.html) it is achieved by keeping the Bluetooth LPM chip active and running autonomously even when the iPhone is switched off.

While all user interaction with the device is turned off, the Bluetooth chip continues to advertise its presence to nearby devices by operating on low-power mode, albeit at intervals larger than the default 15 minutes.

This illustrates that you can never trust a device to be entirely powered off, even when you turn off your phone.

Likewise, the "NoReboot" technique makes it impossible to physically detect if an iPhone is off or not as to all outward appearances your device appears to be shut down.

Furthermore, malware developers and hackers can now gain persistence on iOS devices with this technique, where the usual recommendation of restarting an iPhone to clear infections no longer works.